Automatic Inference and Enforcement of Kernel Data Structure Invariants
Arati Baliga, Vinod Ganapathy and Liviu Iftode

Department of Computer Science, Rutgers University

Gibraltar automatically detects rootkits that violate invariants exhibited by control and non-control data structures in the kernel

Problem Statement

I. Rootkits manipulate kernel data structures to achieve malicious goals
   - Control data: system call table, function pointers
   - Non-control data: virtually any data structure in the kernel

II. Current data integrity monitoring approaches have limitations
   - Automatic monitoring has focussed on control data violations
   - Monitoring non-control data requires manual definition of integrity specifications

Research Challenges

I. Complexity, volume and heterogeneity of kernel data
   - Invariants have to be inferred across several thousands of complex data structures maintained by the kernel

II. Learning Meaningful Invariants
   - Non-control data changes much more frequently than control data.
   - Learning invariants is challenging

Rootkit Detection via Invariant Inference

Observe and infer invariants on kernel data structures

- Invariant Inference is done during the training period
- Inferred invariants are used as integrity specifications of data structure integrity and enforced during normal operation of the system

Invariant Examples

Example 1

poolinfo.tap1 € \{26, 103\}
poolinfo.tap2 € \{20, 76\}
poolinfo.tap3 € \{14, 51\}
poolinfo.tap4 € \{7, 25\}
poolinfo.tap5 == 1

Invariants shown here are violated by the Entropy Pool Contamination Attack [Oakland ‘07]

Example 2

run-list ⊆ all-tasks

Invariant shown here is violated by the Hidden Process Attack [Security ‘06]

Experimental Results

- Detection Accuracy: Tested with 21 rootkits, no false negatives.
- Non-control data attacks

<table>
<thead>
<tr>
<th>Template</th>
<th>Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>var € [a,b,c]</td>
<td>Entropy Pool Contamination</td>
</tr>
<tr>
<td>var € [a,b,c]</td>
<td>Resource Wastage</td>
</tr>
<tr>
<td>len(var) == c</td>
<td>Binary Format</td>
</tr>
<tr>
<td>list1 ⊆ list2</td>
<td>Hidden Process</td>
</tr>
</tbody>
</table>

- False Positives: 0.65%
- Performance Overhead: 0.49%

Invariants Inferred

<table>
<thead>
<tr>
<th>Template</th>
<th>Object</th>
<th>Collection</th>
</tr>
</thead>
<tbody>
<tr>
<td>var € [a,b,c]</td>
<td>643,622</td>
<td>422</td>
</tr>
<tr>
<td>var != 0</td>
<td>49,058</td>
<td>266</td>
</tr>
<tr>
<td>var &gt;0, var &lt; 0</td>
<td>16,696</td>
<td>600</td>
</tr>
<tr>
<td>len(var) == c</td>
<td>NA</td>
<td>4,696</td>
</tr>
<tr>
<td>list1 ⊆ list2</td>
<td>NA</td>
<td>3,580</td>
</tr>
<tr>
<td>Total</td>
<td>709,376</td>
<td>9,564</td>
</tr>
</tbody>
</table>

Implementation

Gibraltar has asynchronous remote access to memory contents of the target

Non-control data offers wider attack surface than control data